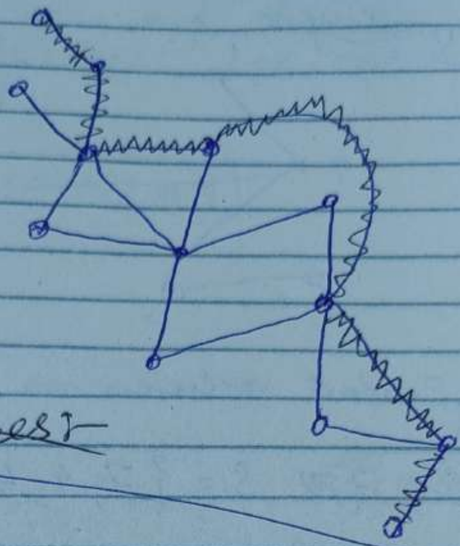
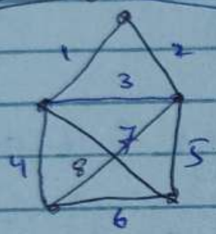
After sessional 1

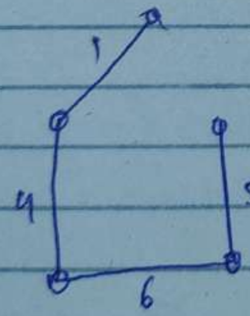
LECTURE # 8

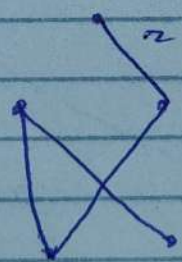


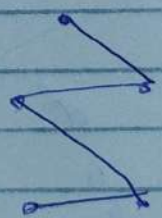
Distance of trees:-



Number of different edges in ~~two~~ two spanning trees of a graph ~~are~~ is called distance of the



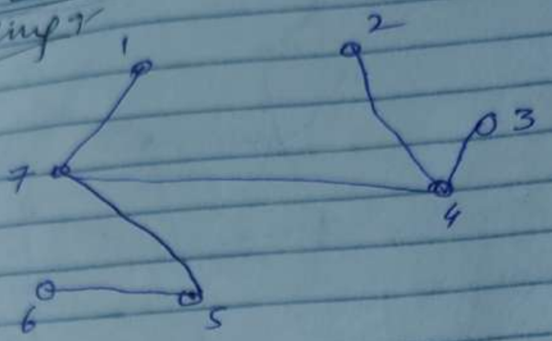




In case of fully connected graph the maximum number of distance of trees could be n-1 (maybe/I think).

PRUFER CODES:-

Encoding:

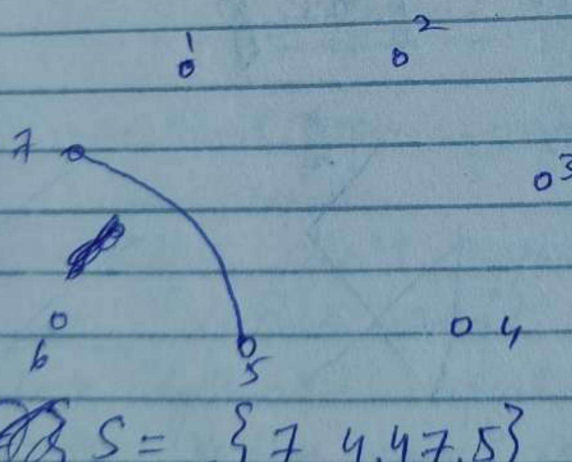


Pendant vertices ⇒ 1, 2, 6, 3

~~S =~~ S = {7, 4, 4, 7, 5}

→ Remove the edge between 1 & 7 and add the 7 in S.
→ Same steps for all pendant vertices.

This is left:

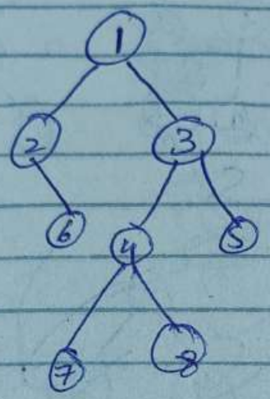


~~S~~ S = {7, 4, 4, 7, 5}
V = {1, 2, 3, 4, 5, 6, 7}

→ Now remove the corresponding element from the start of S & V and connect those vertices in the graph.

→ It is very usefull for sending and recieving tree and also for saving, because it takes less space.

LECTURE #9

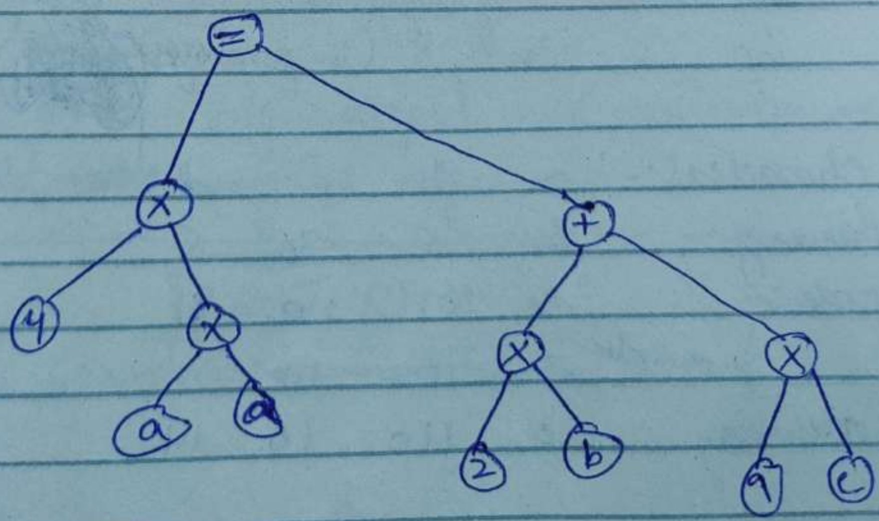


preorder:-

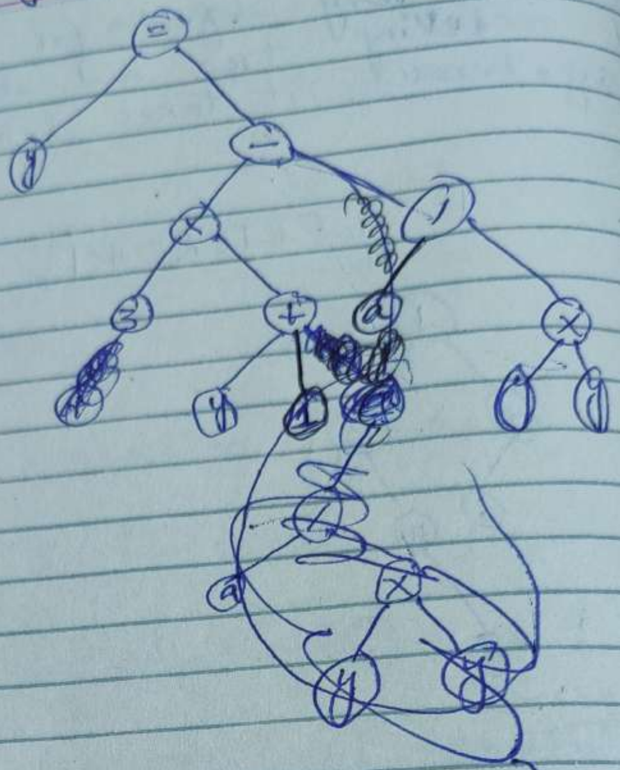
1, 2, 6, 3, 4, 7, 8, 5

→ $4a^2 = 2b + 9c$

$4 \times (a \times a) = (2 \times b) + (9 \times c)$



Preorder:- $y = - \times 3 + y\ 1 / a \times y\ y$



Inorder:-

$y = (3 \times (y + 1)) - (a / (y \times y))$

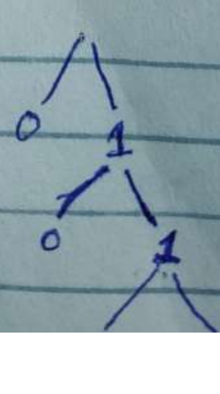
Storage of ~~messages upto~~ characters:-

a, b, c, d

with frequencies

60, 5, 30, 5 (in percentages)

Characters:-	a	b	c	d
frequency:-	60	5	30	5
code:-	00	01	10	11
↓ optimization	~~0~~	~~110~~	~~10~~	
Representation	0	110	10	111



Example:-

Data fi

characters:- a

frequency:- 45

fixed

variable

Assignm

Columns

Airline

departur

and

Minim

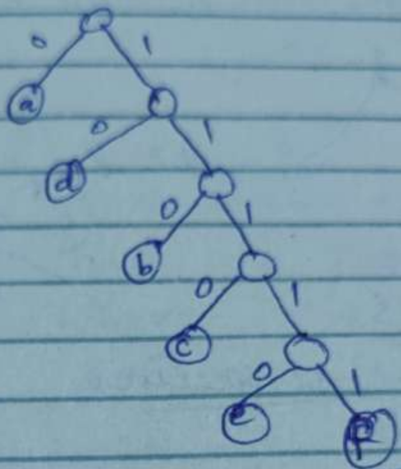
1. P
2. K
3. D

Example:-

Data File: 100,000 characters

characters:-	a	b	c	d	e	f
frequency:-	45	13	12	16	9	5
variable	0	110	1110	10	11110	11111

fixed length code: $3 \times 100,000$
$= 300,000$ bits



Assignment 1 :-

Columns:-

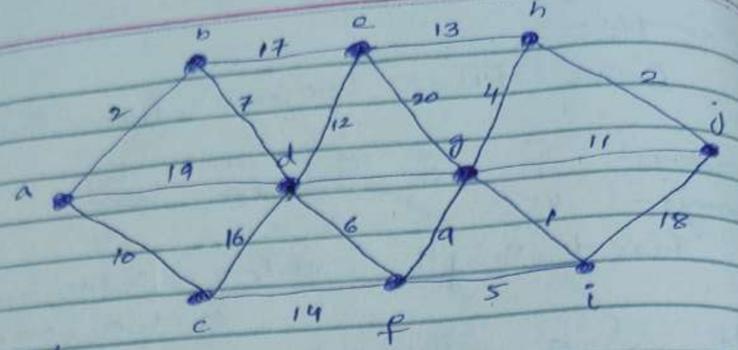
→ Airline, Origin airport, destination airport, departure delay, arrival delay

→ pick origin_airport and destination_airport and add weighted edges.

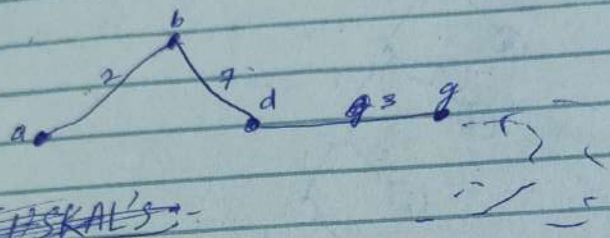
LECTURE # 10

Minimum Spanning Tree:-

1. Prim's
2. Kruskal's
3. Dijekstra's



~~PRIMS~~:-
untraversed = { a, b, c, d, e, f, g, h, i, j }



~~KRUSKAL'S~~:-

PRIMS:-

programming perspectives-

① Traversed nodes: $0 \rightarrow 10$
② Untraversed nodes:- $10 \rightarrow 0$
③ Adjacent edge list:- $e_1 e_2 e_5$, $e_2 e_3 e_6 e_7$

LECTURE # 11

Euler circuit/path/Tour

Hamiltonian path/cycle/circuit

Vertex connectivity:-

Minimum number of vertices must be removed to divide the graph into multiple components.

Q:- This is a planar graph, show it

Lecture # 12

Prim's Algorithm:

We see only the edges that are connected with the edge we are currently at.

Kruskal's Algorithm:-

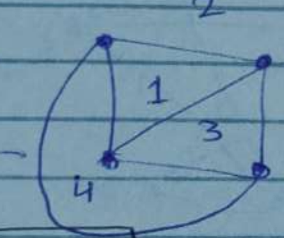
We see the whole graph as a whole and choose the minimum one.

We also check the edge we chosen must not make a cycle.

Planar Graphs:-

Geometric representation of graph in such a way that no two edges intersect each other.

Regions
faces.



Euler' Formula:-

Connected graph with n vertices & e edges has $\boxed{e-n+2}$ regions.

$n = 4$, $e = 6$

Regions $= e-n+2 = 6-4+2$

Regions $= 4$.

To check graph is planar or NOT:-

Graph G is a simple graph with $n > 3$, then the following rule must hold in order for this graph to be planar